

CSE- 1006 LAB Assignment - 4**Academic year:** 2020-2021**Semester:** FALL**Faculty Name:** Dr. Arun kumar Gopu**Date:** 16/3/2022**Student name:** M.Taran**Reg. no.:** 19BCE7346**Exporting data:**

There are plenty of functions for writing data to files. Some of them are:

- Write.table
- Write.csv
- Cat
- writeLines
- Dump
- Dput
- Save
- serialize

EXERCISES

Consider the data set "airquality"

Read first 6 lines into a new data frame "aq" (aq <- head()airquality))

```
> aq <- head(airquality)
> aq
  Ozone solar.R wind Temp Month Day
1   41    190  7.4   67     5    1
2   36    118  8.0   72     5    2
3   12    149 12.6   74     5    3
4   18    313 11.5   62     5    4
5   NA     NA 14.3   56     5    5
6   28     NA 14.9   66     5    6
```

Practice the following exercises on the following data.

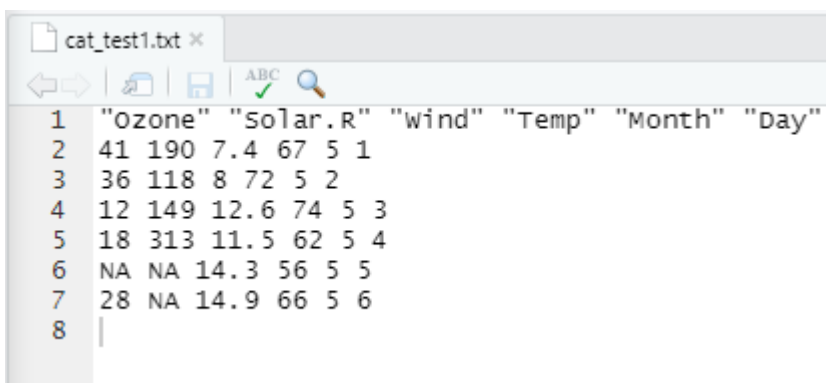
Note: After creating files, open and see the files to comply with the required output.csv files can be opened either by excel or notepad

You can observe the delimiters when you open the file using notepad.

Cat command:

Write a command to export (store/save) data into the file cat_test1.txt (Use only two arguments). After creating the file check the output.

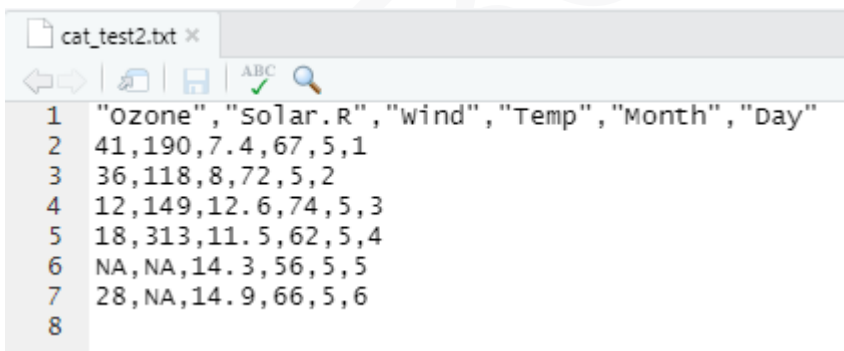
```
> write.table(aq, file = "cat_test1.txt", sep=" ", row.names=FALSE)
```



	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
1	41	190	7.4	67	5	1
2	36	118	8	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6
7						
8						

Write a command to export data into the file cat_test2.txt. Use separator as comma

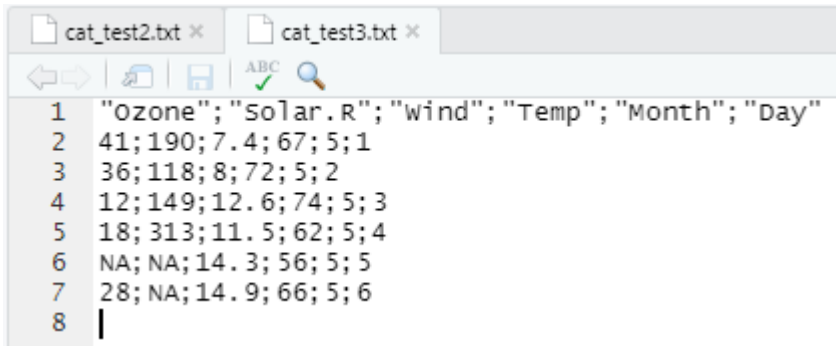
```
> write.table(aq, file = "cat_test2.txt", sep=",", row.names=FALSE)
>
```



	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
1	41,190,7.4,67,5,1					
2	36,118,8,72,5,2					
3	12,149,12.6,74,5,3					
4	18,313,11.5,62,5,4					
5	NA,NA,14.3,56,5,5					
6	28,NA,14.9,66,5,6					
7						
8						

Write a command to export data into the file cat_test3.txt. Use separator as semicolon

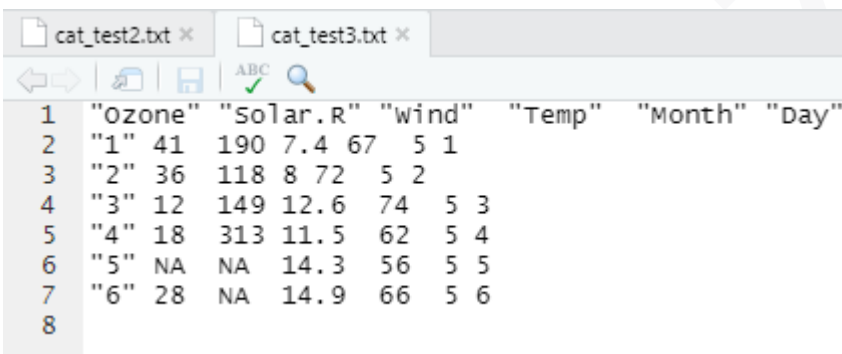
```
> write.table(aq, file = "cat_test3.txt", sep=";", row.names=FALSE)
> |
```



	Ozone	Solar.R	wind	Temp	Month	Day
1	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
2	41	190	7.4	67	5	1
3	36	118	8	72	5	2
4	12	149	12.6	74	5	3
5	18	313	11.5	62	5	4
6	NA	NA	14.3	56	5	5
7	28	NA	14.9	66	5	6
8						

Write a command to export data into the file cat_test3.txt. Use separator as tab (use \t to insert tab)

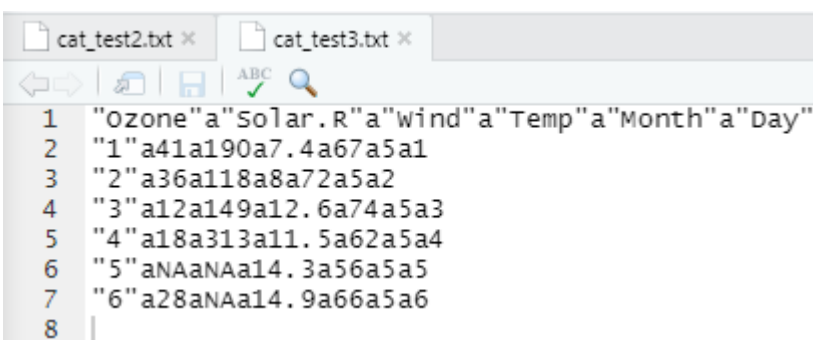
```
> # Export a data frame to a text file
> write.table(aq,
+           file = "cat_test3.txt",
+           sep = "\t",
+           row.names = TRUE,
+           col.names = TRUE)
```



	Ozone	Solar.R	wind	Temp	Month	Day
1	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
2	"1"	41	190	7.4	67	5 1
3	"2"	36	118	8	72	5 2
4	"3"	12	149	12.6	74	5 3
5	"4"	18	313	11.5	62	5 4
6	"5"	NA	NA	14.3	56	5 5
7	"6"	28	NA	14.9	66	5 6
8						

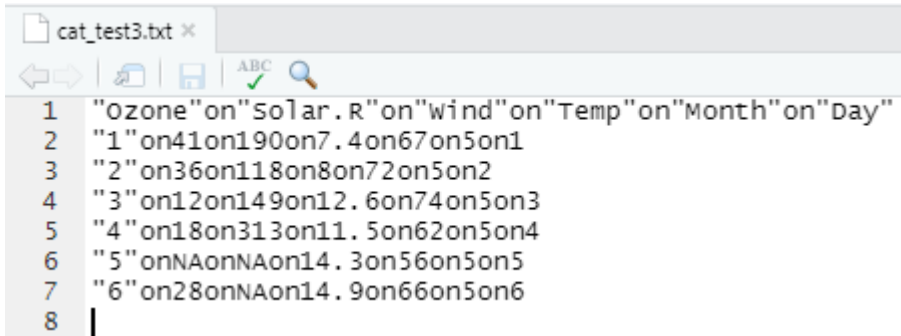
Can a separator be any string that you want to insert as delim?? Experiment with it.

```
> write.table(aq,
+           file = "cat_test3.txt",
+           sep = "a",
+           row.names = TRUE,
+           col.names = TRUE)
```



	Ozone	Solar.R	wind	Temp	Month	Day
1	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
2	"1"	a41a	190a	7.4a	67a	5a1
3	"2"	a36a	118a	8a	72a	5a2
4	"3"	a12a	149a	12.6a	74a	5a3
5	"4"	a18a	313a	11.5a	62a	5a4
6	"5"	aNAa	NAa	14.3a	56a	5a5
7	"6"	a28a	NAa	14.9a	66a	5a6
8						

```
> write.table(aq,  
+           file = "cat_test3.txt",  
+           sep = "on",  
+           row.names = TRUE,  
+           col.names = TRUE)
```



```
1 "Ozone"on"Solar.R"on"wind"on"Temp"on"Month"on"Day"  
2 "1"on41on19on7.4on67on5on1  
3 "2"on36on118on8on72on5on2  
4 "3"on12on149on12.6on74on5on3  
5 "4"on18on313on11.5on62on5on4  
6 "5"onNAonNAon14.3on56on5on5  
7 "6"on28onNAon14.9on66on5on6  
8 |
```

What are Delimiters? What is the need of delimiters?

Programming languages use delimiters in different coding scenarios to determine specific type and instruction boundaries. Because delimiters - such as commas and full stops - define different condition types, the delimiter concept is very similar to the English language.

What is the difference between over writing and appending?

If Append is set to true, records are appended to the file. If Append is set to false, the file is overwritten.

Write a command to append the same data into the file cat_test1.txt. After creating the file check the output.

```
write.table(aq,'cat_test1.txt',append=TRUE)
```

Write a command to append the same data into the file cat_test2.txt. Use separator as comma

```
write.table(aq,'cat_test2.txt',sep=",",append=TRUE)
```

Write a command to append the same data into the file cat_test3.txt. Use separator as semicolon

```
write.table(aq,'cat_test3.txt',sep=";",append=TRUE)
```

Write a command to append the same data into the file cat_test3.txt. Use separator as tab (use \t to insert tab)

```
write.table(aq,'cat_test3.txt',sep="\t",append=TRUE)
```

Can you store the data into .csv files using cat command?

No, The strength of the cat() command is in being able to join items together, this allows

to use it to make output messages in custom commands and scripts.

Write a command to export the data to cat_test1.csv.

```
write.table(aq,file='cat_test1.csv')
```

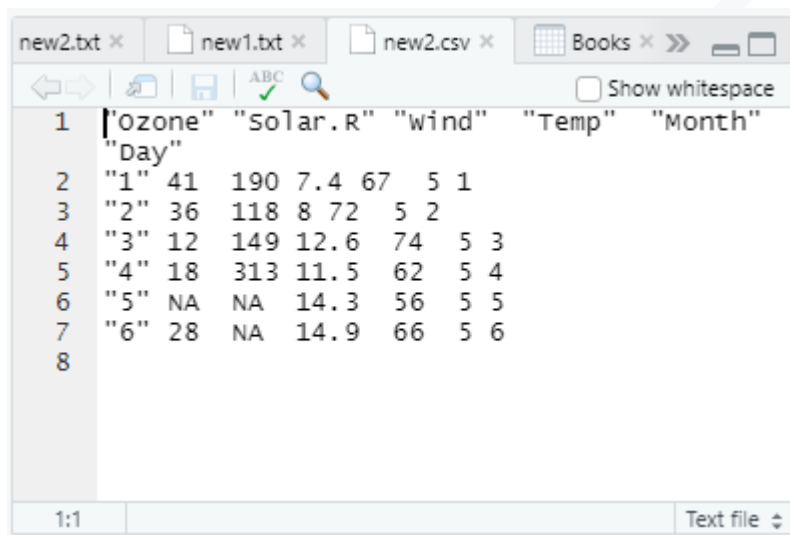
Write a command to export the data using various separators such as semicolon and tab and store them in new .csv files. Open newly created files using both excel and notepad. Observe the difference.

Using csv:

```
write.table(aq,file='new.csv',sep=",")
```

```
write.table(aq,file='new1.csv',sep=";")
```

```
write.table(aq,file='new2.csv',sep="\t")
```

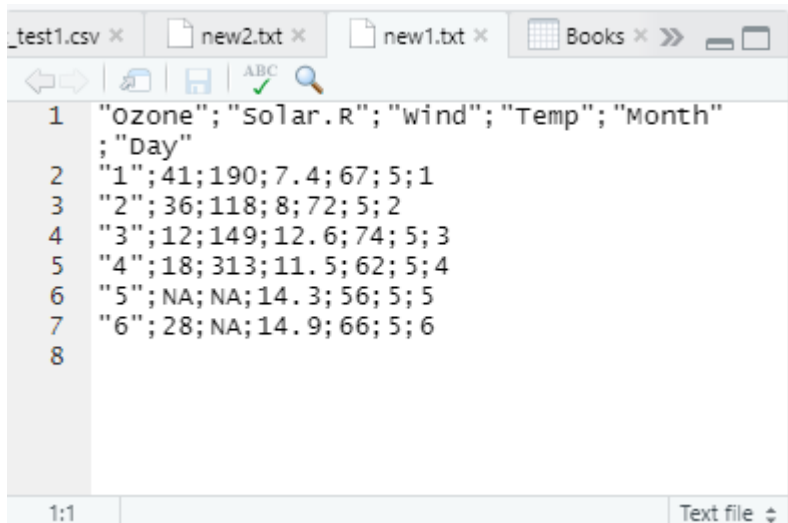


	"Ozone"	"Solar.R"	"wind"	"Temp"	"Month"	"Day"
1	"1"	41	190	7.4	67	5 1
2	"2"	36	118	8 72	5	2
3	"3"	12	149	12.6	74	5 3
4	"4"	18	313	11.5	62	5 4
5	"5"	NA	NA	14.3	56	5 5
6	"6"	28	NA	14.9	66	5 6
7						
8						

Using Notepad:

```
write.table(aq,file='new.txt',sep=",")
```

```
write.table(aq,file='new1.txt',sep=";")
```

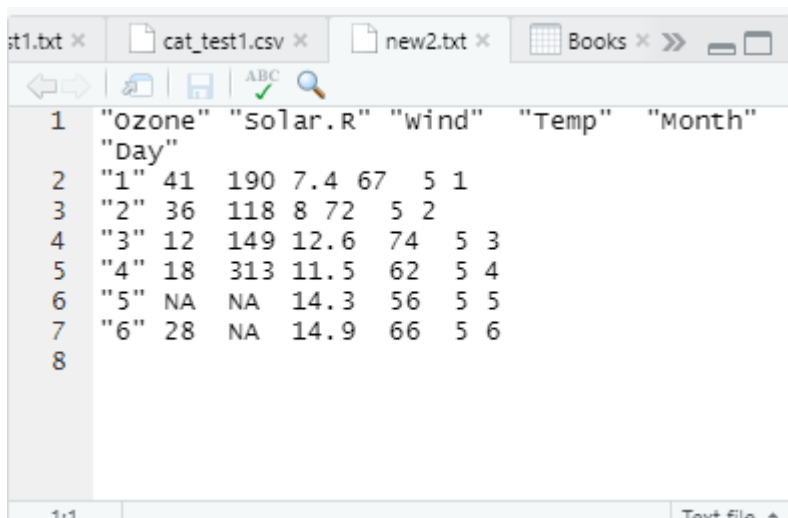


```

1 "Ozone"; "Solar.R"; "wind"; "Temp"; "Month"
  ; "Day"
2 "1"; 41; 190; 7.4; 67; 5; 1
3 "2"; 36; 118; 8; 72; 5; 2
4 "3"; 12; 149; 12.6; 74; 5; 3
5 "4"; 18; 313; 11.5; 62; 5; 4
6 "5"; NA; NA; 14.3; 56; 5; 5
7 "6"; 28; NA; 14.9; 66; 5; 6
8

```

```
write.table(aq,file='new2.txt',sep="\t")
```



```

1 "Ozone" "Solar.R" "wind" "Temp" "Month"
  "Day"
2 "1" 41 190 7.4 67 5 1
3 "2" 36 118 8 72 5 2
4 "3" 12 149 12.6 74 5 3
5 "4" 18 313 11.5 62 5 4
6 "5" NA NA 14.3 56 5 5
7 "6" 28 NA 14.9 66 5 6
8

```

What are other arguments you can use while exporting data to a file using cat function. (Hint: Use ?cat command to learn about other arguments and experiment on them)

There are a total of six arguments in the cat function to export data to a file.

They are datafile, sep all, labels, apped. cat(... file="", sep=" ", fill=FALSE,labels=NULL, append=FALSE)

write.table command:

Repeat the above exercises using the write.table command. Here, file extension is .txt as write.table stores data into text files.

write.csv command:

Repeat the above exercises using the write.csv command. Here, file extension is .csv as write.csv stores data into comma separated files.

Other commands:

Practice the below commands using the examples given in lecture slides

- writeLines, readLines
- dump, source
- dput, dget
- save, load
- serialize, unserialize
- Scan

Importing data

There are a few principal functions reading data into R.

read.table, read.csv, for reading tabular data readLines, for reading lines of a text file

source, for reading in R code files (inverse of dump)

dget, for reading in R code files (inverse of dput)

load, for reading in saved workspaces

unserialize, for reading single R objects in binary form

read.delim(), for reading data separated by tab

-----EXERCISES-----

Note:

read.csv is identical to read.table except that the default separator is a comma. There are variants in the functions in read.csv and read.delim such as read.csv2 and read.delim2.

read.table command :

Write a command to read the data from cat_test1.txt. Display the output.

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test1.txt")
```

```
read.table("cat_test1.txt")
```

Write a command to read the data from cat_test2.txt. Display the output. Modify your command to get the output as given below

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test2.txt")
```

```
read.table("cat_test2.txt")
```

Write a command to read the data from remaining files and display the output as in the above figure.

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test3.txt", sep=" ")
```

```
read.table("cat_test3.txt")
```

Write a command to read and display the data without quotes on strings

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test4.txt", sep="\t")
```

```
read.table("cat_test4.txt")
```

```
> aq=head(airquality)
> write.table(aq,file = 'cat_test4.txt', sep='\t')
> read.table('cat_test4.txt')
  Ozone Solar.R wind Temp Month Day
1    41     190  7.4   67     5    1
2    36     118  8.0   72     5    2
3    12     149 12.6   74     5    3
4    18     313 11.5   62     5    4
5     NA      NA 14.3   56     5    5
6    28      NA 14.9   66     5    6
>
```

Write a command to read the data without column names

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test4.txt", col.names = FALSE)
```



```
read.table("cat_test4.txt")
```

```
> aq=head(airquality)
> write.table(aq,file = 'cat_test4.txt', col.names = FALSE)
> read.table('cat_test4.txt')
  V1 V2  V3   V4 V5 V6 V7
1  1  1 41 190  7.4 67  5  1
2  2  2 36 118  8.0 72  5  2
3  3  3 12 149 12.6 74  5  3
4  4  4 18 313 11.5 62  5  4
5  5  5 NA  NA 14.3 56  5  5
6  6  6 28  NA 14.9 66  5  6
>
```

Write a command to read the data without row names

```
aq=head(airquality)
```

```
write.table(aq,file = "cat_test4.txt", row.names = FALSE)
```

```
read.table("cat_test4.txt")
```

```
> aq=head(airquality)
> write.table(aq,file = 'cat_test4.txt', row.names = FALSE)
> read.table('cat_test4.txt')
    V1    V2    V3    V4    V5    V6
1 Ozone solar.R wind Temp Month Day
2   41    190   7.4   67     5     1
3   36    118    8   72     5     2
4   12    149  12.6   74     5     3
5   18    313  11.5   62     5     4
6 <NA>   <NA>  14.3   56     5     5
7   28    <NA>  14.9   66     5     6
>
```

Read.csv command:

Repeat the above exercises using read.csv command

```
aq=head(airquality)
```

```
write.csv(aq,file = "cat_test1.csv")
```

```
read.csv("cat_test1.csv")
```

```
> aq=head(airquality)
> write.csv(aq,file = 'cat_test1.csv')
> read.csv('cat_test1.csv')
  X Ozone Solar.R wind Temp Month Day
1 1   41    190  7.4   67     5    1
2 2   36    118  8.0   72     5    2
3 3   12    149 12.6   74     5    3
4 4   18    313 11.5   62     5    4
5 5   NA     NA 14.3   56     5    5
6 6   28     NA 14.9   66     5    6
```

```
aq=head(airquality)
```

```
write.csv(aq,file = "cat_test2.csv", row.names = FALSE)
```

```
read.csv("cat_test2.csv")
```

```
> aq=head(airquality)
> write.csv(aq,file = 'cat_test2.csv', row.names = FALSE)
> read.csv('cat_test2.csv')
  Ozone Solar.R wind Temp Month Day
1   41    190  7.4   67     5    1
2   36    118  8.0   72     5    2
3   12    149 12.6   74     5    3
4   18    313 11.5   62     5    4
5   NA     NA 14.3   56     5    5
6   28     NA 14.9   66     5    6
```